

Solução da lista 6

Cristiano de Carvalho

Questão 1

Crie uma função que receba os dados e a confiança e retorne o intervalo de confiança para a média. Use a função: $IC_{(\mu, 1-\alpha)} = \bar{X} \pm t_{\frac{\alpha}{2}; n-1} \times \frac{s}{\sqrt{n}}$, sendo $t_{n-1; \frac{\alpha}{2}}$ o valor apropriado da tabela t para o seu nível de significância desejado e seus graus de liberdade. Dica: use a função $qt()$ para achar o valor e use o nível de confiança para calcular.

```
funQ1<-function(dados, confianca = 0.95){  
  #Formula do intervalo de confianca  
  media <- mean(dados)  
  liberdade <- length(dados)  
  alfa <- 1-confianca  
  estatistica <- qt(1-(alfa/2), df=(liberdade-1)) *(sd(dados)/sqrt(liberdade))  
  intervaloupper <- media+estatistica  
  intervalolower <- media-estatistica  
  intconf <- c(intervalolower, intervaloupper)  
  return(intconf)  
}
```

Questão 2

Crie uma função que receba um valor predeterminado e retorne 1 caso ele esteja dentro do intervalo qualquer e 0 caso contrário.

```
funQ2<-function(intervalo, valorteste){  
  #Funcao responsavel por testar os se o valor teste esta dentro do intervalo  
  if(intervalo[1]<=valorteste & intervalo[2]>valorteste){  
    f<-1  
  }  
  else f<-0  
  return(f)  
}
```

Questão 3

Usando a função replicate, gere, a partir de uma normal qualquer, 1000 bancos de dados de tamanho 1000 cada. Após isso, calcule qual a porcentagem de amostras que gera intervalos de confiança que contém o valor da média teórica.

```
mu=40  
sd=14  
n=1000  
  
funQ3 <-function(j){
```

```

#Calcula a media do vetor e com isso consegue calcular a taxa de acertos
acertos <- mean(j)
erros <- 1 - acertos
cat("A taxa de acertos foi: ", (acertos), "\n", "A taxa de erros foi: ", (erros), "\n")
}

replicador<-replicate(n, funQ2(funQ1(rnorm(100, mu, sd)), valorteste=mu), simplify=TRUE)
#Gera 1000 bancos e testa eles usando o valor teste como o mu

funQ3(replicador)

## A taxa de acertos foi: 0.944
## A taxa de erros foi: 0.056

```

Questão 4

Transforme a questão anterior em uma função única e utilize o sapply para fazer o teste com diferentes tamanhos de amostra.

```

funQ4<-function(mu, sd, n, amostra){
  #funcao da questao 3 de forma direta
  testando <- replicate(n, funQ2(funQ1(rnorm(100, mu, sd)),valorteste=mu), simplify=TRUE)
  acertos <- (sum(testando)/n)*100
}

sapply(c(15, 40, 100, 1000), funQ4, n=100000, mu=50, sd=1, simplify = F)

## [[1]]
## [1] 94.99
##
## [[2]]
## [1] 94.987
##
## [[3]]
## [1] 95.041
##
## [[4]]
## [1] 95.003

```

Questão 5

Agora, utilizando o mapply, faça o teste para diferentes valores de amostras e variâncias.

```

mapply(funQ4, mu=c(40, 10, 15),sd=c(10, 40, 30), n=100, amostra=1000)

## [1] 97 91 97

```

Questão 6

Resolva a integral da função $f(x) = x^{1/2} - x^2, 0 \leq x \leq 1$ por Monte Carlo e faça um gráfico mostrando o comportamento para diferentes tamanhos de amostras.

```

#Valor correto
funQ6_1<- function(x){
  y=((x**(1/2))-(x**2))
  return(y)
}

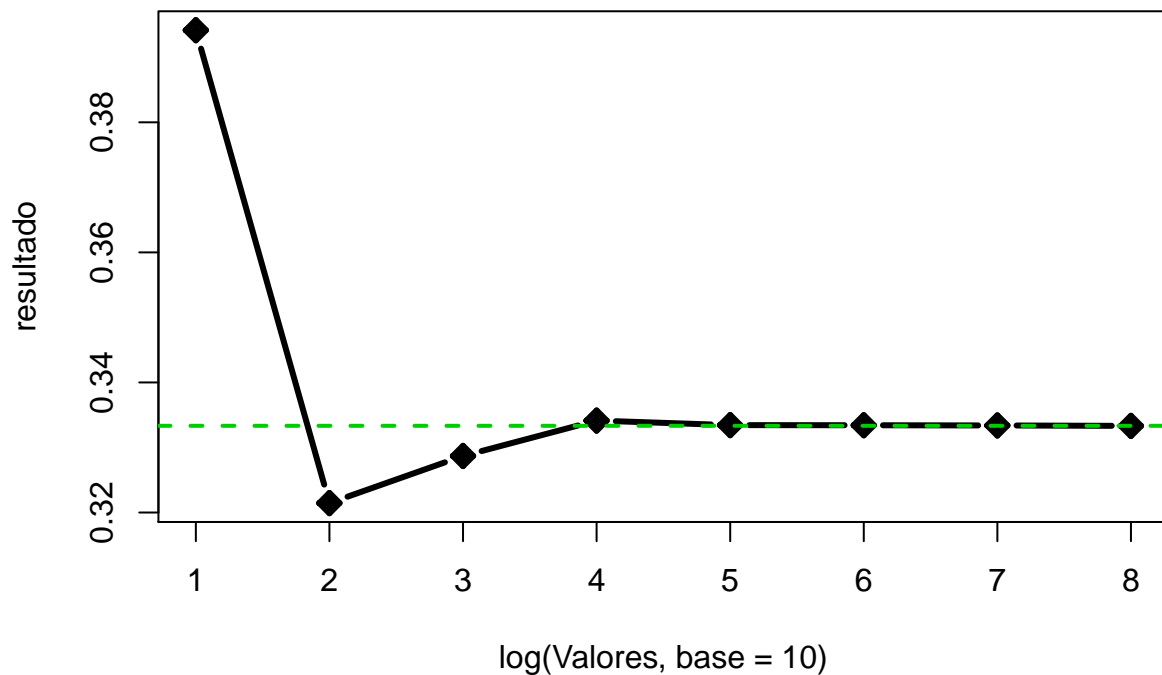
confiavel<-integrate(f = funQ6_1, lower = 0, upper = 1)$value

#Monte Carlo
funQ6_2<-function(Y){
  x = runif(Y, 0, 1)
  return(mean((x**(1/2))-(x**2)))
}

Valores<- 10**(1:8)
resultado <- sapply(X= Valores, FUN = funQ6_2)

plot(log(Valores,base = 10), resultado, type = "b", pch = 9, lwd = 3)
abline(h=confiavel, col = 3, lty = 2, lwd = 2)

```



Questão 7

Seja N o tamanho amostral, faça uma função que gere uma amostra com 10 valores de N a partir de uma distribuição de Poisson com média igual 8. Para cada valor de N , gere um vetor z_1, \dots, z_N , em que cada Z_i é gerado de uma distribuição Weibull com parâmetros 2 e 1. Obtenha o valor de Y dado pelo mínimo de Z_1

até Z_N . Retorne tudo em uma lista com 10 elementos, sendo que cada elemento é uma sublista contendo o vetor de Z e o valor Y .

```
funQ7_aux<-function(N){
  amostra = rweibull(N, shape = 2, scale = 1)
  minimo = min(amostra)
  return(list(amostra = amostra, minimo = minimo))
}

FunQ7 <- function(M = 10){
  Ns <- rpois(M, 8)
  sapply(Ns, funQ7_aux, simplify = F)
}

FunQ7(10)
```

```
## [[1]]
## [[1]]$amostra
## [1] 0.3192990 0.1522766 1.0861699 1.0633123 0.8790262 0.3562724 0.5503852
##
## [[1]]$minimo
## [1] 0.1522766
##
##
## [[2]]
## [[2]]$amostra
## [1] 1.1461965 1.3004104 0.7904694 1.0111491 0.8815295 1.5611229
##
## [[2]]$minimo
## [1] 0.7904694
##
##
## [[3]]
## [[3]]$amostra
## [1] 1.01047552 0.92463046 0.80601406 1.06989790 0.82749649 0.54287853
## [7] 0.65469783 0.08980972 1.34049508 0.30060998 1.06787212
##
## [[3]]$minimo
## [1] 0.08980972
##
##
## [[4]]
## [[4]]$amostra
## [1] 0.4910537 0.3842060 0.7344866 1.3330402 1.0712134
##
## [[4]]$minimo
## [1] 0.384206
##
##
## [[5]]
## [[5]]$amostra
## [1] 1.7116784 0.4349267 0.6714737 0.4715514 0.5310354 0.5450282 1.5958703
##
## [[5]]$minimo
## [1] 0.4349267
```

```
##
##
## [[6]]
## [[6]]$amostra
## [1] 0.9193663 2.3669123 1.1996150 0.7724623 1.1475974 0.9264229 1.0743772
## [8] 0.9451769
##
## [[6]]$minimo
## [1] 0.7724623
##
##
## [[7]]
## [[7]]$amostra
## [1] 0.03494556 1.38147570 0.62324633 0.23680420 1.62774687 1.88156232
## [7] 1.13522870 0.73131825 1.68250242 1.23146457 0.75244753 0.40704298
##
## [[7]]$minimo
## [1] 0.03494556
##
##
## [[8]]
## [[8]]$amostra
## [1] 1.1511064 0.8707266 1.4273099 0.5155304 1.3062666 1.4236803 0.7858487
## [8] 1.7975385 1.3022759 0.5041840 0.5253225 0.6769026
##
## [[8]]$minimo
## [1] 0.504184
##
##
## [[9]]
## [[9]]$amostra
## [1] 0.6157029 1.6512583 1.3163376 1.0263316 0.7811063 1.0815450 0.3704977
## [8] 1.1092186 1.4868721 0.8590614 0.5756264
##
## [[9]]$minimo
## [1] 0.3704977
##
##
## [[10]]
## [[10]]$amostra
## [1] 0.4593990 0.9115431 0.2570446 0.8067517 0.5483874 0.9328983 0.2578454
## [8] 0.6426909
##
## [[10]]$minimo
## [1] 0.2570446
```

Questão 8

Defina uma nova classe para um vetor numérico de sua escolha e, para essa nova classe, crie um novo método da função `mean` que retorna a média do vetor após eliminar o valor máximo e o valor mínimo.

```
mean.Q8<-function(x){
  x<-sort(x)
  n<-length(x)
```

```

    mean(x[-c(1, n)])
}

vetor<-runif(7, min = 1, max = 5)
mean(vetor)

## [1] 3.027453

class(vetor) = 'Q8'
mean(vetor)

## [1] 3.049576

```

Questão 9

Crie um operador que multiplica o valor de duas matrizes e, depois, calcula a soma dos elementos da matriz resultante.

```

"%**%"<-function(A, B){
  mult = A**B
  soma = sum(mult)
  return(soma)
}

M1 = matrix(1:9, 3)
M2 = matrix(9:17, 3, byrow = F)
"%**%"(M1, M2)

## [1] 1809

M1**M2

## [1] 1809

```